



**HAL**  
open science

## Energy-driven Consolidation in Digital Home

Rémi Druilhe, Matthieu Anne, Jacques Poulou, Laurence Duchien, Lionel Seinturier

► **To cite this version:**

Rémi Druilhe, Matthieu Anne, Jacques Poulou, Laurence Duchien, Lionel Seinturier. Energy-driven Consolidation in Digital Home. 28th ACM Symposium on Applied Computing (SAC'13), 1st Software Engineering Aspects of Green Computing (SEAGC) Track, Mar 2013, Coimbra, Portugal. pp.1157-1162. hal-00762626

**HAL Id: hal-00762626**

**<https://inria.hal.science/hal-00762626>**

Submitted on 7 Dec 2012

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Energy-driven Consolidation in Digital Home

Remi Druilhe\*†, Matthieu Anne\*, Jacques Pulou\*  
Laurence Duchien†, Lionel Seinturier†

\*Orange Labs  
28 Chemin du Vieux Chêne  
38420 Meylan, France  
firstname.lastname@orange.com

†INRIA Lille-Nord Europe  
LIFL (UMR CNRS 8022)  
Université Lille 1, France  
firstname.lastname@inria.fr

## ABSTRACT

As our society becomes more and more digital, the corresponding demand for electric energy is increasing. Despite the power efficient design of devices, this rising trend of energy consumption does not weaken because of more and more devices used in our daily life. Collaboration strategies between devices can reduce their overall electrical consumption. *Consolidation* – *i.e.*, migrating tasks among devices to place into low power state or to switch off a maximum of unused devices – is a mean of optimizing the consumption of a group of devices. So far, consolidation is mainly used in datacenters. Here, we propose a model to extend this approach to Digital Home. This model takes into account properties, such as the unforeseeable appearance of devices or restrictions due to task nature. Its implementation in a Digital Home environment saves around 25% of the energy consumption in a scenario based on the daily life of a family of four persons.

## Categories and Subject Descriptors

C.4.5 [Performance of Systems]: [Performance Attributes];  
G.1.1 [Optimization]: [Constrained Optimization]

## General Terms

Performance, Algorithms

## Keywords

Energy, Digital Home, Consolidation, Quality of Service, Event-processing

## 1. INTRODUCTION

The growing electrical power consumption is one of the major problem of our decade [18]. The digitalization of our society contributes to this growing power demand. IT domains such as datacenter, office computing or Digital Home

(DH) see their pool of devices growing constantly, increasing the power consumption. For example, a study of the energy consumption of U.S.A households shows that the percentage of IT devices, *e.g.*, laptop, smartphone, set-top box (STB), and appliances power consumption tends to increase from 17% in 1978 to 31% in 2005 [1]. Moreover with the growing pervasive computing in the DH [4], new tasks are introduced using devices that are potentially multifunction, all the more increasing energy consumption.

Actions are being taken to reduce the power consumption of IT devices. The main approach consists in either switching off unused devices or putting them into low power state [2]. *Consolidation* enables to optimize this approach, migrating existing tasks on a subset of devices to maximize the number of unused devices [12]. Currently, consolidation is mainly used in datacenters but other environments such as DH could benefit from this approach to reduce its energy consumption.

The challenge is to define a consolidation model addressing DH to reduce energy consumption. Our proposition enables consolidation to take into account three properties issued from DH: (a) dynamicity (*e.g.*, appearance of a device); (b) heterogeneity (*e.g.*, power consumption, hardware resources); (c) Quality of Service (QoS) requirements.

We implement this model in a constraint solver using appropriate objective function and constraints. The implementation considers task portability over devices as a pre-requirement for our solution. Our solution is assessed by an experiment deployed using a simulated DH. Our results demonstrate a 25% reduction in the energy consumed by a family of four in our digital home scenario.

This paper is organized as follows: Section 2 gives details about adopted models for each property described. Section 3 addresses the implementation of our model through a constraint solver, evaluates our solution and discusses results. Section 4 compares to related work in energy consumption. Finally, Section 5 concludes this paper.

## 2. MODELLING OF THE ENVIRONMENT

*Consolidation* permits to optimize tasks placement among a datacenter in order to reduce energy consumption [12]. We take this approach and apply it to the DH through a dedicated model that consider three properties. This section introduces general terms and defines these properties. An optimization workflow is also defined to represent our model.

### 2.1 Properties Considered

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SAC'13 March 18-22, 2013, Coimbra, Portugal.

Copyright 2013 ACM 978-1-4503-1656-9/13/03 ...\$15.00.

We define three properties from DH environment: (a) dynamicity, (b) heterogeneities and (c) QoS. First, dynamicity of the DH is addressed, *i.e.*, unforeseeable appearance or disappearance of devices or tasks. For example, mobile devices such as a smartphone regularly leave and come back to the network. Dynamicity makes the consolidation deployment out of date and constantly forces a new optimization.

We identify three kinds of heterogeneity: (a) power consumption, (b) device hardware resources and (c) task hardware requirements. In the DH, we consider that devices do not consume the same power and do not provide the same hardware resources, in the same quantity. Tasks are different and do not require the same resources and not in the same quantity. For example, a video player requires a screen contrary to an antivirus and they both do not require the same quantity of CPU resources.

To address QoS, *i.e.*, the collective effect of service and performances that determine the degree of satisfaction of the service [19], from both user and system points of view, we subdivide it into two aspects. We first consider the Quantity of Resources (QoR) to address performance degradation of task from the system point of view. For example, a task needs enough hardware resources to execute in a non-degraded mode. Secondly we consider the mobility of a task from the user point of view, *i.e.*, the need for a task to stay on a device because it is being used. For example, a browser interacting with the user cannot migrate without the consent of the user whereas a download task can, as the interaction with the user, is lower.

Properties of dynamicity, heterogeneity and QoS are the challenges addressed and our solution includes them through the model explained in this section.

## 2.2 Basic Terms

We consider that a *system*, noted  $s$ , is composed of a finite set of devices. A *device*, noted  $d$ , corresponds to an entity able to run one or several tasks concurrently. A *task*, noted  $t$ , is a work, including required data to process, to be run on a device and is not essential to the proper functioning of the device (*e.g.*, *init* is essential for the proper functioning of an device under Linux whereas *wget* is not). We note the set of devices  $D$  and the set of tasks  $T$  executing in the system. A system is defined as a tuple:  $s = \langle D, T \rangle$ .

A device  $d$  is composed of *resources*  $r$  provided in  $Q_d^r$  quantities. A device uses those resources to run tasks. A task  $t$  requires Quantities of Resources (QoR)  $Q_t^r$  to be executed on a device with the required QoS. For example, if a datacenter is a system, a device is a server, a task is a program and resources of a device correspond to a processor and its RAM. Quantities of resources are defined: 1000 MIPS for a processor and 500 GB for RAM. The use of a common benchmark on each device permits to assume that common metrics can be used for resources on different devices. For example, it permits to assume that a Intel Dual Core of 1000 MIPS has the same capabilities as a ARM 9 of 1000 MIPS.

## 2.3 Events

In an environment such as the DH, mobile device, *e.g.*, smartphone, may frequently appear or disappear from the network. Others, *e.g.*, STB, desktop computer, rarely leave the network. Tasks are more volatile and start or stop depending on the habits of the household. These unpredictable changes in the system represent the dynamicity. The dy-

namicity is handled through events happening in the system. An event is relative to a device or a task and may modify the set of devices,  $D$ , or the set of tasks,  $T$ , that the system has to deal with. We discern the system state before event  $\tau$  and the system state after event  $\tau + 1$ . Hereafter is the list of events we consider:

- Appearance of a device  $d$ :  $D^{\tau+1} = D^\tau \cup \{d\}$
- Disappearance on a device  $d$ :  $D^{\tau+1} = D^\tau \setminus \{d\}$
- Start of a task  $t$ :  $T^{\tau+1} = T^\tau \cup \{t\}$
- End of a task  $t$ :  $T^{\tau+1} = T^\tau \setminus \{t\}$

## 2.4 Organization Plan

The *organization plan* makes the correspondence, at a given time, between tasks and devices executing these tasks. We note  $T_d^\tau$ , the set of tasks running on device  $d$  at the time  $\tau$ . This plan is represented through a matrix  $A^\tau = (a_{dt}^\tau)_{1 \leq d \leq Card(D^\tau), 1 \leq t \leq Card(T^\tau)}$  which is given at the time  $\tau$ . To fulfil the matrix  $A^\tau$ , we proceed as follows:

$$a_{dt}^\tau = \begin{cases} 1 & \text{if } t \in T_d^\tau \\ 0 & \text{else} \end{cases} \quad (1)$$

Matrix  $A^\tau$  is dynamically updated and depends on the evolution of  $D$  and  $T$ . When an event occurs, it changes one of these sets, triggering an update of the matrix  $A^\tau$  to  $A^{\tau+1}$ . This update drives the system to optimize itself.

## 2.5 Resources and QoR

A task can run if the resource required for the task is available on the device.

### *QoR of a device.*

The available QoR on a device  $d$  is the set of QoR of each of its resources  $r$ . We note  $Q_d$  the QoR associated to a device and  $Q_d^r$  the one associated to a specific resource  $r$  (*e.g.*, CPU, RAM):

$$Q_d = \begin{pmatrix} Q_d^{cpu} \\ Q_d^{ram} \\ \dots \\ Q_d^r \end{pmatrix} \quad (2)$$

For example for the CPU, it can correspond to 1000 MIPS – 1 MIPS being the unity of use of a CPU – and for the RAM to 500 GB – 1 MB being the unity of use of RAM. This QoR is maximal when no task is running on it:  $T_d^\tau = \emptyset$ .

### *QoR of a task.*

The QoR required by a task is defined using same metric as defined for a device. It can differ from considered data. For example, a video player does not require same QoR for two distinct movies. In this case, we consider two distinct tasks including a common video player but two different movies. Furthermore, the QoR required by a task can change depending on task's life cycle [14]. For example, a video player does not require the same QoR for two different stages of a movie. Finally the QoR required for a task depends also on its implementation. Finding the QoR required for a task is not an easy way, that's why we do not address this problem in this paper. In our case, we only consider the worst case scenario, *i.e.*, the maximal QoR required for a task. We consider this QoR known and constant throughout all task life cycle. The required QoR of a task  $t$ ,  $Q_t$ , is defined in the same way as  $Q_d$ :

$$Q_t = \begin{pmatrix} Q_t^{cpu} \\ Q_t^{ram} \\ \dots \\ Q_t^r \end{pmatrix} \quad (3)$$

## 2.6 Power consumption of devices

The devices can present different active states and even different Low Power (LP) states [10, 15]. In the sequel, we consider only an unique active state and an unique LP state for each device. States of a device are noted  $S_d$ . We define: (a) a state where it is able to execute tasks (*active state*), *on*, and (b) a state where it is not able to execute tasks (*low power state*), *lp*. Therefore, in our model, each device is either in the active state or in the LP state.

$P_d$  denotes the consumption of device  $d$ . In our model, power consumption can take two values corresponding to the LP and active states of the device: respectively  $P_d^{lp}$ , which is constant, and  $P_d^{on}$ , which depends on the load of the device. We assume that  $P_d^{on}$  is a function of the resources consumed by the hosted tasks at the time  $\tau$  [6]. Currently, because devices do not provide same hardware resources, there is no common model for all the devices. Current models of power consumption of devices are complex in order to represent faithfully the power consumption of each resource [16]. We abstract from the consumption of devices and define the consumption of an device  $d$  as follows:

$$P_d = \begin{cases} P_d^{lp} & \text{if } S_d = lp \\ P_d^{on}(Q_d^\tau) & \text{if } S_d = on \end{cases} \quad (4)$$

The power consumption of the system is the addition of the power consumption of each device at the time  $\tau$  (cf. Equation 5). If a device does not hosts any task, we consider that it should be in low power state.

$$P_s^\tau = \sum_{d \in D^\tau} (\delta_d^\tau \times P_d^{on}(Q_d^\tau) + (1 - \delta_d^\tau) \times P_d^{lp}) \quad (5)$$

$\delta_d^\tau$  corresponds to the activity of an device  $d$  at the time  $\tau$ . It is equal to 1 if the device hosts at least one task, 0 else (cf. Equation 6).

$$\delta_d^\tau = 1 - \prod_{t \in T^\tau} (1 - a_{dt}^\tau) \quad (6)$$

## 2.7 Energy consumption of migration

Consolidation introduces an additional cost to consider in the optimization process: the energy consumption of task migrations. The migration process involves two devices: the device  $d$  from which migration starts and the device  $d'$  at which migration ends. A task migration process can be decomposed into three steps, with different durations: (a)  $\tau_s$  to save the state of the task on device  $d$ , (b)  $\tau_t$  to transfer the task and its state from device  $d$  to device  $d'$ , and (c)  $\tau_r$  to restore the state of the task on device  $d'$ . Equation 7 defines the energy consumption of the migration process.

$$E_m^\tau = \sum_{t \in T^\tau} \gamma_t \times (P_d^{on} \times (\tau_s + \tau_t) + P_{d'}^{on} \times (\tau_t + \tau_r)) \quad (7)$$

$\gamma_t$  corresponds to the migration of a task  $t$  in the system. It is equal to 1 if the task appears to be migrated by the optimization function, 0 else.

Mobility	Example
$\{L, S\}$	Watching a video
$\{L, M\}$	<i>Follow me</i> task [7]
$\{A, S\}$	Files concatenation
$\{A, M\}$	Downloading

Table 1: Spatial mobility examples

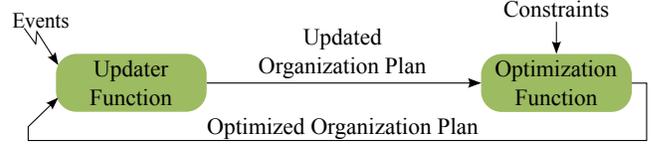


Figure 1: Optimization workflow.

## 2.8 Mobility

To handle the QoS from the user point of view, we define the *mobility* of a task. This mobility is *spatial* (*i.e.*, moving a task onto another device of the system, choice of the deployment of the task) or *temporal* (*i.e.*, delaying the execution of a task). We only address the spatial mobility here, and leave the temporal one for future work.

We define the spatial mobility using two properties: (a) the choice of the device to execute the task, the *deployment* and (b) the ability for a task to be moved onto another device of the system, the *migrability*. The *deployment* can take two values: localized (L) or alocalized (A). A localized task is a task that must be deployed onto a specific device. This is the case for the tasks strongly interacting with the users (*e.g.*, watching a video, browsing the web). An alocalized task is a task that can be deployed on any device. For example, downloading a file from the web can be done on several devices.

The *migrability* can also take two values: migratable (M) and static (S). A migratable task is a task that can be moved from one device to another during runtime (*e.g.*, music player, downloading a file from the web). On the contrary, a static task is executed on an unique device and cannot be moved (*e.g.*, antivirus), thus this device remains in active state. An example of mobility of tasks is given in Table 1.

## 2.9 Optimization workflow

An optimization workflow, illustrated in Figure 1, represents the executive model. At the initial state, we consider the organisation plan as being optimized. The optimization workflow starts when an event occurs. Events, described in subsection 2.3, modify the organization plan  $A^\tau$  (cf. subsection 2.4) through the *updater function*. This function just apply events to the set of devices or tasks. It produces an *updated organization plan*  $A^{\tau+1}$  which might not be power efficient. To optimize the system, this updated organization plan is given as an input to the *optimization function*. The optimization function finds the best power efficient organization plan: the *optimized organization plan*  $A'^{\tau+1}$ . It based on constraints and an objective function focusing on minimizing power consumption.

This workflow addresses properties of dynamicity through events to update itself. It addresses heterogeneities and QoS through constraints defined in the optimization function. Its

Devices	Active state (W)	Low power state (W)	CPU (BogoMIPS)	RAM (Mo)	ROM (Mo)	NIC (kb/s)	Mic	Speaker	Screen
DECT	2 – 3	1	50	10	0	50	1	1	0
Desktop computer	80 – 128	5	5000	4000	250000	1000000	0	0	1
Smartphone	1 – 4	1	1500	11	113	270000	1	1	1
STB	20 – 30	3	3000	1000	300000	200000	0	0	0
TV	100 – 110	1	3000	2000	3000	270000	0	1	1

**Table 2: Informations on devices**

Tasks	Mobility	CPU (BogoMIPS)	RAM (Mo)	ROM (Mo)	NIC (kb/s)	Mic	Speaker	Screen	Start – End
Call	$\{L, M\}$	50	10	0	50	1	1	0	18:00 – 00:00
Download	$\{A, M\}$	300	2	3000	500	0	0	0	18:15 – 22:25
Web browsing	$\{L, S\}$	590	257	30	50	0	0	1	17:05 – 19:15 & 22:45 – 23:00
Surveillance	$\{A, M\}$	1000	100	10	500	0	0	0	16:00 – 00:00
Video	$\{L, S\}$	1540	133	10	200	0	0	1	17:00 – 19:15 & 20:00 – 23:00

**Table 3: Informations on tasks**

evaluation is detailed through the next section.

### 3. VALIDATION

DH fits the properties considered. Dynamicity because devices leave or come back regularly from DH network. Heterogeneities because devices are different (*e.g.*, smartphone, laptop, STB). And QoS because users expect tasks to be available and adapted to their needs. From our model in the previous section, we define the objective function and constraints used in a constraint solver. The evaluation is made using a realistic 8 hours scenario in the DH.

#### 3.1 Implementation issues

The implementation of our optimization process is done through a *Constraint Satisfaction Problem* (CSP). A CSP is a finite sequence of variables with respective domains together with a finite set of constraints, each on a subsequence of these variables [3]. In this section, we define the expression of the power consumption of a device. We then provide the expressions of the objective function and constraints for each property we address.

##### *Power consumption model.*

We take a formula based on a CPU as model for power consumption of a device. The CPU is a resource present on each devices. It is also the most power consuming resource on the majority of devices subject to different workloads [6]. For our validation, we choose a linear model based on CPU load:

$$P_d^{on}(Q_d^{cpu\ used}) = (P_{d_{max}} - P_{d_{min}}) \times \frac{Q_d^{cpu\ used}}{Q_d^{cpu\ total}} + P_{d_{min}} \quad (8)$$

$P_{d_{max}}$  is the power consumption of the device when its overall resources are in use.  $P_{d_{min}}$  corresponds to the power consumption when no task is executing on the device  $d$ .

##### *Objective function.*

Finding the *organization plan optimized* consists in finding the organization plan that minimizes the power consumption of the system among possible organization plans satisfying each constraint. The power consumption of the system is given by Equation 5. Equation 9 is the objective function that minimize the power consumption of the system. This function seeks an optimum solution.

$$\min(P_s) = \min\left(\sum_{d \in D^\tau} (\delta_d^\tau \times P_d^{on} + (1 - \delta_d^\tau) \times P_d^{lp})\right) \quad (9)$$

##### *Constraint on power consumption.*

An organization plan is considered as candidate for becoming the optimized organization plan  $A'^\tau$  if it consumes less energy than the updated organization plan  $A^\tau$  considering the energy consumption for migration  $E_m^\tau$ :

$$P_{A'^\tau} \times \Delta\tau + E_m^\tau < P_{A^\tau} \times \Delta\tau \quad (10)$$

where  $P_{A^\tau}$  is the power consumption of the updated organization plan and  $P_{A'^\tau}$  is the power consumption of the optimized organization plan.  $\Delta\tau$  is the duration between two events. Note that, if  $\Delta\tau$  tends to 0, migration is useless because the migration tends to consume more than the system. On the other hand, if  $\Delta\tau$  tends to  $\infty$ , the migration decision is always taken because energy consumption of the migration is negligible. In our case, we consider that the system knows the duration between two events.

##### *Constraint on resources and QoR.*

We make the hypothesis that the subtraction of the QoR required by task ( $Q_t$ ) from QoR available on a device ( $Q_d$ ) permits to determine if the task can run on this device. We also consider that the system knows how much resources are available on a device after the deployment of a task. This is given by the Equation 11.

$$Q_d(T_d^\tau \cup t) = Q_d(T_d^\tau) - Q_t \quad (11)$$

In order to guarantee QoS, no value of  $Q_d(T_d^T \cup t)$  should be negative for any resources. If it is, the task cannot be deployed on this device without a degradation of QoS. Moreover, if a required resource for a task is missing on the device, the task cannot be deployed on this device.

When a task disappears, QoR of task  $t$  is released from the QoR of the device  $d$ :

$$Q_d(T_d^T) = Q_d(T_d^T \cup t) + Q_t \quad (12)$$

### Constraint on tasks mobility.

The mobility property permits to define four types of tasks. Table 1 exposes each combination with an example and permits to define constraints on mobility of tasks: if  $t$  is *static* and is included in  $T_d^T$ , then  $a_{dt}^T = 1$ . This constraint imposes to let device  $d$  in active state (setting to 1  $a_{dt}^T$ ) if the task  $t$  is static and is already deployed on the device  $d$ .

## 3.2 Evaluation

To evaluate our solution, we define a typical scenario which happens regularly in the DH. We consider a family of two children and two adults. Children come back from school at 17:00. They start watching TV (*Video* task) and surfing the web (*Web Browsing* task) on a desktop computer. Adults come back at 18:00. A child starts to download a big file (probably a game or a movie) at 18:15 (*Download* task). Then, the family goes to dinner at 19:15 and finishes at 20:00 for watching news and evening movie. At 22:25, the initiated download finishes and a child checks the file at 22:45. At 23:00, everyone goes to sleep.

Among other tasks, the *Surveillance* task tracks intruders using a camera. Image processing is executed locally, within the DH and must be available 24/7. Image processing can be deployed in various devices, this task is considered as alocalized and migratable. The *Call* task corresponds to the ability of a person to be reachable through any phone. This task can be migrated to a device which has a MIC and a speaker, such as a DECT or a smartphone. We consider tasks migration through a transfer of both a code file and a state file, permitting to resume the task after migration. We assume the total size of this set around 10 MB. The LAN bandwidth is around 200 Mb/s.

Devices and tasks used in the scenario are described in Table 2 and Table 3, respectively. From these data, constraints for the model are provided. Mobility of tasks in Table 3 refers to mobility subsection 2.8. Values come from real devices and tasks we have monitored to extract those data. We choose to implement this scenario in a simulator based on Choco, a Java-based constraint solver [11].

Given this scenario, we plot the energy consumption graph with and without any optimization. Dotted curve on Figure 2 shows power consumption without optimization. Solid curve shows power consumption with optimization. We notice that during the first hour, the only task to be executed on the STB is the surveillance task. When TV and desktop computer are active, the surveillance task is migrated to TV and STB can be shutdown, resulting in a saving of 23 W. Change in power consumption are not impacted by the start of the download task, around 18:15. At dinner time (19:15), TV and desktop computer are unused, *i.e.*, no more video task nor web browsing task, our solution decides to migrate surveillance and download task on the STB to shutdown the

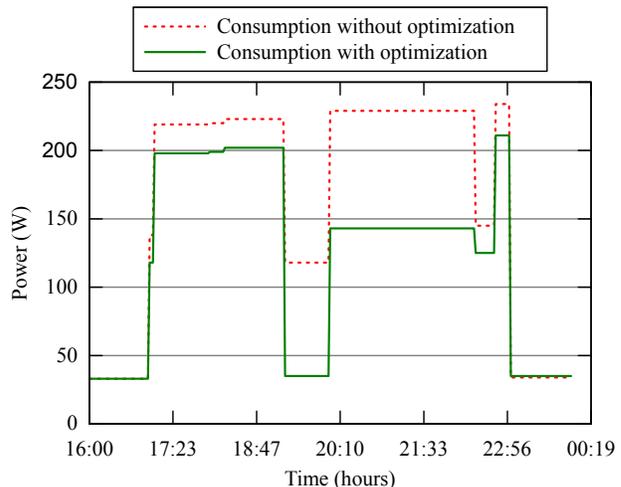


Figure 2: Energy consumption with and without optimization.

desktop computer. The difference between both curves gives a reduction of the energy consumption about 25% in this 8 hours scenario.

Consolidation is known as a bin packing problem [12]. We try to find the smallest set of devices able to run the larger set of tasks. Despite this is a NP hard problem, our model take in the order of seconds to process. This is due to the lower set of devices and tasks present in a DH in contrary to a datacenter which holds of much more devices and tasks.

## 4. RELATED WORK

Among the literature addressing similar properties in a consolidation approach, [17, 8] focus on heterogeneity of power consumption and heterogeneity of quantities of resources of devices using a model for some hardware components (*e.g.*, CPU, NIC). Applied to a datacenter, these models reduce the energy consumption but in comparison to our approach they only consider QoS from the system point of view. Moreover, they do not deal with dynamicity.

Among related works relying on QoS from the user point of view, solutions consist in monitoring user events (*e.g.*, leaving computer) and then applying policies to migrate tasks to dedicated devices [5, 21]. Contrary to our approach, these solutions imply that dedicated servers never suffer from a lack of hardware resources. Furthermore, they do not consider the varying mobility of tasks.

In a P2P approach, *i.e.*, no dedicated servers, tasks are deployed among multiple subsystems, sharing hardware resources [9]. But a user can refuse to share hardware resources, even if there is enough to run other tasks, limiting possible optimizations. Moreover it does not consider event-driven power optimizations which is an important property we consider in our approach.

Finally, separating rendering and computation (*e.g.* first is at home, latter in a datacenter) permits to consider mobility of tasks and placement of them to reduce the overall power consumption without losing QoS from user point of view [20]. In comparison to our approach, this model does not consider exotic hardware resources (*e.g.*, webcams, Bluetooth) that some tasks may require and are locals resources. Moreover it is not always interesting, from energy point of

view, to move computation to a remote system [13].

## 5. CONCLUSION

This paper proposes the use of a consolidation approach to reduce energetic consumption of IT devices. We introduce the dynamics of the system, its heterogeneity and the quality of service. We define a generic model and implement it in a real-life scenario in the DH. It demonstrates that a gain is possible permitting to apply the approach of consolidation in an heterogeneous and dynamic environment.

Therefore, mobility of tasks implies to rely on the habits of users and devices present in the system to reduce their energy consumption. For example, in our scenario, if we replace the desktop computer by a laptop, a smaller gain is expected because of the energy consumption of the laptop. Nevertheless, depending on the scenario, a reduction of the consumption of IT devices is always possible. Furthermore, this approach is not limited to DH. It could be extended to office computing that owns similar properties.

This paper does not take into account the uncertainty of the estimation of the quantity of resources required by a task. This QoR may varies depending on the device executing the task and the life cycle of the task. For future work, finding suitable resources for a task may provide a fine-grained optimization. Moreover, in our validation the period between events is known whereas events can be unforeseeable. Learning from users habits could bring further reduction because the system could estimate the duration between two events. Additionally, delaying low priority tasks could modify our model, permitting to integrate the temporal mobility and bringing further energy reduction. Finally, for now, this model has not been deployed in a realistic environment. We do not know how many energy it consumes. It is part of a future work to deploy our solution and consider its own consumption.

## 6. ACKNOWLEDGMENTS

This work is partially supported by the French FUI project EconHome and by the French Ministry of Higher Education and Research, Nord-Pas de Calais Regional Council and FEDER through the Contrat de Projets Etat Region Campus Intelligence Ambiante (CPER-CIA) 2007-2013.

## 7. REFERENCES

- [1] U. E. I. Administration. Share of energy used by appliances and consumer electronics increases in u.s. homes. <http://www.eia.gov/consumption/residential/reports/2009/electronics.cfm>, 2012.
- [2] Y. Agarwal, S. Savage, and R. Gupta. Sleepserver: A software-only approach for reducing the energy consumption of pcs within enterprise environments. In *USENIX Annual Technical Conference*. USENIX Association, 2010.
- [3] K. Apt. *Principles of constraint programming*. Cambridge Univ Pr, 2003.
- [4] A. Bottaro, A. Gérodolle, and P. Lalanda. Pervasive service composition in the home network. In *Advanced Information Networking and Applications*. IEEE, 2007.
- [5] T. Das, P. Padala, V. Padmanabhan, R. Ramjee, and K. Shin. Litegreen: Saving energy in networked desktops using virtualization. In *USENIX annual technical conference*. USENIX Association, 2010.
- [6] D. Economou, S. Rivoire, C. Kozyrakis, and P. Ranganathan. Full-system power analysis and modeling for server environments. In *Workshop on Modeling, Benchmarking, and Simulation*. Citeseer, 2006.
- [7] A. Harter, A. Hopper, P. Steggle, A. Ward, and P. Webster. The anatomy of a context-aware application. In *International conference on Mobile computing and networking*. ACM, 1999.
- [8] T. Heath, B. Diniz, E. Carrera, W. Meira Jr, and R. Bianchini. Energy conservation in heterogeneous server clusters. In *Symposium on Principles and practice of parallel programming*. ACM, 2005.
- [9] H. Hlavacs, R. Weidlich, K. Hummel, A. Houyou, A. Berl, and H. de Meer. Distributed energy efficiency in future home environments. *Annals of Telecommunications*, 63(9):473–485, 2008.
- [10] M. Josefiok. Towards an energy abstraction layer. *Energy-Efficient Applications*, pages 25–30, 2012.
- [11] N. Jussien, G. Rochart, X. Lorca, et al. Choco: an open source java constraint programming library. In *Workshop on Open-Source Software for Integer and Constraint Programming*, 2008.
- [12] G. Khanna, K. Beaty, G. Kar, and A. Kochut. Application performance management in virtualized server environments. In *Network Operations and Management Symposium*. IEEE, 2006.
- [13] K. Kumar and Y. Lu. Cloud computing for mobile users: Can offloading computation save energy? *Computer*, 43(4):51–56, 2010.
- [14] M. Louvel, J. Pulous, A. Plantec, and J. Babau. Quantity of resource aggregation for heterogeneous resource reservation for multimedia applications. In *Emerging Technologies and Factory Automation*. IEEE, 2010.
- [15] F. Maker and Y. Chan. A survey on android vs. linux. *University of California*, pages 1–10, 2009.
- [16] J. McCullough, Y. Agarwal, J. Chandrashekar, S. Kuppaswamy, A. Snoeren, and R. Gupta. Evaluating the effectiveness of model-based power characterization. In *USENIX Annual Technical Conf*. USENIX Association, 2011.
- [17] R. Nathuji, C. Isci, and E. Gorbato. Exploiting platform heterogeneity for power efficient data centers. In *International Conference on Autonomic Computing*. IEEE, 2007.
- [18] U. Nations. 2012 international year of sustainable energy for all. <http://www.un.org/en/events/sustainableenergyforall/>, 2012.
- [19] I. Recommendation. *800: Terms and definitions related to quality of service and network performance including dependability*, 1994.
- [20] W. Vereecken, L. Deboosere, P. Simoens, B. Vermeulen, D. Colle, C. Develder, M. Pickavet, B. Dhoedt, and P. Demeester. Power efficiency of thin clients. *European Transactions on Telecommunications*, 21(6):479–490, 2010.
- [21] W. Zhong, G. Shi, Z. Zhao, and F. Xia. Parasite: A system for energy saving with performance improvement in networked desktops. In *Green Computing and Communications*. IEEE, 2011.